

Тернопільський національний технічний
університет імені Івана Пулюя

Кафедра автоматизації
технологічних процесів
і виробництв

Лабораторна робота № 3
з курсу
”Мікропроцесорні та програмні
засоби автоматизації”

Дослідження структури та
програмування ОМЕОМ МК51
з використанням програмного
емулятора
Single-Chip Machine 2.2

Тернопіль 2018

Методичні вказівки до лабораторної роботи №3 “Дослідження структури та програмування ОМЕОМ МК51 з використанням програмного емулятора Single-Chip Machine 2.2” з курсу «Мікропроцесорні та програмні засоби автоматизації». Медвідь В.Р., Пісьціо В.П., - Тернопіль: ТНТУ, 2018 - 24 с.

Відповідальні за випуск

доцент, к.т.н. Медвідь В.Р.,

асистент Пісьціо В.П.

Для студентів напрямку: 151 "Автоматизація та комп'ютерно-інтегровані технології"

Лабораторна робота №3

Дослідження структури та програмування ОМЕОМ МК51 з використанням програмного емулятора Single-Chip Machine 2.2

Мета роботи:

Ознайомлення з структурою та програмуванням мікро-ЕОМ сімейства MCS-51 на прикладі ОМЕОМ і80С51.

1. Загальні відомості про сімейство MCS51

Однокристалні мікроЕОМ (ОМЕОМ) сімейства МК51 (MCS51) відносять до пристроїв з 8-розрядною організацією. Розроблені по n-МОН технології, вони зберегли свою архітектуру при переході на технологію КМОН, що дозволило на порядок знизити енергоспоживання, зберігши повну наступність апаратних і програмних засобів. В даний час серійно випускають лише ОМЕОМ, виконані за КМОН технологією.

Базовий мікроконтролер сімейства і80С51 має такі характеристики:

- об'єм резидентної пам'яті програм – 4 кбайт;
- тип резидентної пам'яті програм – репрограмоване ПЗП РПЗП;
- об'єм резидентної пам'яті даних – 128 байт;
- мінімальна частота тактових сигналів 1,2 МГц;
- максимальна частота тактових сигналів 12 МГц;
- напруга живлення, $+5\text{ В} \pm 10\%$;
- струм споживання, 8 мА;
- обсяг зовнішньої адресованої пам'яті програм, 64 кбайт;
- обсяг зовнішньої адресованої пам'яті даних, 64 кбайт;
- система команд ОМЕОМ і80С51 містить 111 базових команд;
- багаторівнева система переривань підтримує переривання від п'яти джерел.

Мікросхема має два зовнішніх входи запитів переривань.

Особливість і80С51 – обробка бітових даних, що дозволяє використовувати бінарну логіку, котра оперує бітами внутрішнього ОЗП та регістрів. Ця особливість широко використовується у промисловій автоматизації. Ще одна корисна особливість – чотири незалежні набори регістрів, які дуже значно зменшують затримки при обслуговуванні переривань.

До складу ОМЕОМ і80С51 входять такі додаткові пристрої:

- 4 восьмирозрядні паралельні порти вводу-виводу;
- два 16-розрядні лічильники-таймери;
- послідовний порт;
- тактовий генератор;
- блок регістрів спеціальних функцій;
- система захисту програм від несанкціонованого доступу.

У позначенні ОМЕОМ МК-51 цифри мають таке призначення:

- перша цифра показує розрядність АЛП (8 біт);
- друга цифра задає тип внутрішньої пам'яті програм: 0 – ПЗП масочного типу, 3 – ПЗП з зниженою напругою живлення (+3В); 7 – репрограмоване з ультрафіолетовим стиранням, 9 – репрограмоване ПЗП з електричним стиранням,
- літера указує технологію виготовлення: С – пристрій, виконаний за КМОН технологією, відсутність літери – указує n-МОН технологію;
- останні дві цифри задають код сімейства: 51 – МК51 (MCS51), 31 – варіант без внутрішнього ПЗП програм.

Якщо перед зазначеним позначенням є літери, то вони означають конструктивне виконання:

- D – керамічний корпус DIP 40 виводів;
- P – пластиковий корпус DIP 40 виводів;
- N – корпус PLCC, 44 виводи.

Керамічний корпус DIP має вбудоване вікно, закрите кварцовим склом, для стирання ультрафіолетовим випромінюванням записаної в пам'ять програм.

Пластиковий корпус DIP вікна не має, тому ОМЕОМ Р87С51 вважається однократно програмованим виробом, призначеним для дрібносерійного виробництва.

Випускають також мікросхеми з підвищеною швидкодією – із граничним значенням тактової частоти 16, 20 і 24 МГц.

2. Умовне позначення та призначення виводів однокристалльної мікроЕОМ сімейства MCS51

Умовне позначення контролера показано на рис.1.

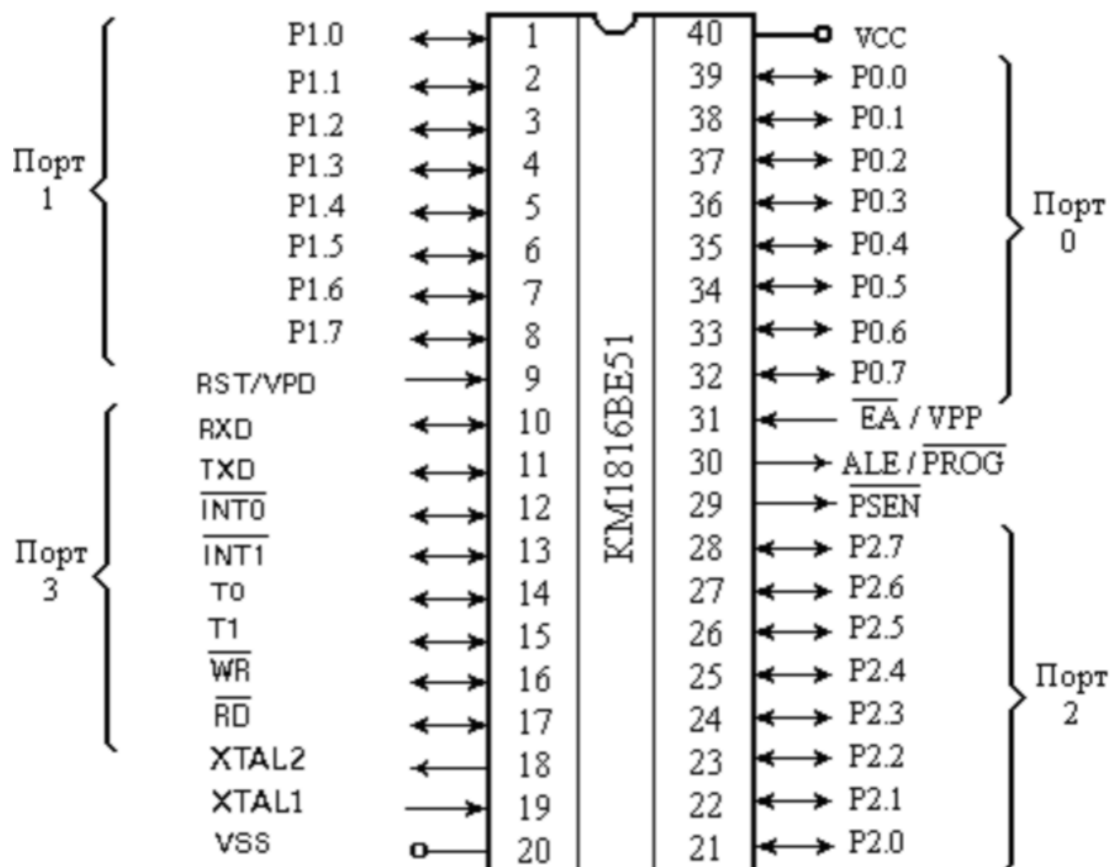


Рис. 1. Позначення та найменування виводів MCS51

Призначення виводів мікросхеми подано в таблиці 1.

Табл. 1. Призначення виводів KM1816BE51

Позначення	Номер виводу	Призначення лінії
1	2	3
P1.0-P1.7	1-8	Порт вводу-виводу 1.
RST/VPD	9	Сигнал скидання мікросхеми/напруга програмування внутрішнього ПЗП. Логічний 0 дозволяє нормальну роботу мікросхеми, логічна 1 протягом 2 машинних циклів переводить мікросхему в режим скидання, рівень напруги 12 В на виводі призводить до програмування мікросхеми.
P3.0/RxD	10	Лінія 0 порту вводу-виводу P3 може працювати як вхід даних в послідовному коді.
P3.1/TxD	11	Лінія 1 порту вводу-виводу P3 може працювати як вихід даних в послідовному коді.

Табл. 1. Призначення виводів KM1816BE51

Позначення	Номер виводу	Призначення лінії
1	2	3
P3.2/ $\overline{\text{INT0}}$	12	Лінія 2 порту вводу-виводу P3 може працювати як вхід запиту переривання або вхід дозволу лічильника 0
P3.3/ $\overline{\text{INT1}}$	13	Лінія 3 порту вводу-виводу P3 може працювати як вхід запиту переривання або вхід дозволу лічильника 1
P3.4/T0	14	Лінія 4 порту вводу-виводу P3 може працювати як вхід відліку внутрішнього лічильника 0.
P3.5/T1	15	Лінія 5 порту вводу-виводу P3 може працювати як вхід відліку внутрішнього лічильника 1.
P3.6/ $\overline{\text{WR}}$	16	Лінія 6 порту вводу-виводу P3 може працювати як сигнал запису у зовнішній ЗП даних або порт вводу-виводу.
P3.7/ $\overline{\text{RD}}$	17	Лінія 7 порту вводу-виводу P3 може працювати як сигнал читання зовнішнього ЗП даних, або порта вводу-виводу.
XTAL2	18	Вивід для підключення зовнішнього резонатора
XTAL1	19	Вивід для підключення зовнішнього резонатора
Vss	20	Земля (напруга 0В).
P2.0/(A8)- P2.7/(A16)	21-28	Порт вводу-виводу P2, лінії порту також використовуються при адресації зовнішньої пам'яті та пристроїв вводу-виводу для виводу старшого байту адреси.
$\overline{\text{PSEN}}$	29	Сигнал читання зовнішньої пам'яті програм. Логічний 0 на лінії вказує, що відбувається читання зовнішньої пам'яті програм.
$\overline{\text{ALE/}}$ $\overline{\text{PROG}}$	30	При нормальній роботі це вихід сигналу ALE - стробуючого сигналу адреси. Перехід сигналу зі стану лог. 1 в стан лог. 0 вказує на наявність адреси на лініях порту P0. В режимі програмування лінія $\overline{\text{PROG}}$ є входом, на який поступає імпульс програмування
$\overline{\text{EA /VPP}}$	31	Вхід заборони роботи внутрішньої пам'яті програм. Логічна 1 на вході дозволяє звертання до внутрішньої пам'яті програм. При логічному 0 на вході при звертанні за адресами, що відповідають внутрішній пам'яті програм, звертання відбувається до зовнішньої пам'яті програм.
P0.0-P0.7	39-32	Порт вводу-виводу P0, лінії порту також використовуються при звертанні до зовнішньої пам'яті програм та даних, а також до пристроїв вводу-виводу для виводу молодшого байта адреси і вводу-виводу даних.
Vcc	40	Напруга живлення 5В.

Всі лінії портів вводу-виводу P1-P3 є квазідвонаправленими і можуть працювати як входи і виходи загального призначення, зрозуміло, з певними обмеженнями.

3. Структурна схема мікросхеми

На відміну від більшості універсальних мікропроцесорів однокристальна мікро-ЕОМ сімейства MCS51 побудована на основі *Гарвардської архітектури*, тобто має розділені пам'ять даних та програм.

Основу структурної схеми MCS51 (рис. 2) складає внутрішня двонаправлена 8-бітна шина, яка зв'язує між собою всі основні вузли та пристрої мікросхеми: резидентну пам'ять програм (РПП), резидентну пам'ять даних (РПД), АЛП з схемою десяткової корекції, блоки регістрів спеціальних функцій, пристрій керування, два лічильники - таймери, блок переривань і порти вводу/виводу.

Розглянемо призначення кожного з вказаних блоків.

3.3. Слово стану програми

В табл. 2 наводиться перелік ознак слова стану програми (ССП), даються їхні символічні імена та описуються умови їх формування.

Табл. 2. Формат слова стану програми (ССП)

Символ	Позиція	Ім'я та призначення
C	PSW.7	Ознака перенесення
AC	PSW.6	Ознака допоміжного перенесення
F0	PSW.5	Ознака користувача 0
RS1	PSW.4	Вибір банку регістрів. Встановлюється і скидається програмою для вибору
RS0	PSW.3	робочого банку регістрів
OV	PSW.2	Ознака переповнення
–	PSW.1	Не використовується
P	PSW.0	Ознака паритету

Найбільш використовуваним ознакою ССП є ознака перенесення (C), що приймає участь і модифікується в процесі виконання багатьох операцій, включаючи додавання, віднімання і зсув. Ознака (C) встановлюється, якщо в старшому біті результату виникло перенесення або позичання. При виконанні множення і ділення ознака C скидається.

Ознака переповнення (OV) фіксує арифметичне переповнення при операціях над цілими числами зі знаком і робить можливим використання арифметики в додаткових кодах. Ознака OV встановлюється, якщо результат операції додавання/віднімання не вміщається в семи бітах і старший (восьмий) біт результату не може інтерпретуватися як знаковий. При виконанні операції ділення ознака OV скидається, а у випадку ділення на нуль встановлюється. При множенні ознака OV встановлюється, якщо результат більший 255.

Ознака AC встановлюється у випадку, якщо при виконанні операції додавання/віднімання між тетрадами байту виникло перенесення/позичання. Встановлюється і скидається при виконанні команд додавання і віднімання і сигналізує про перенесення з третього розряду в четвертий.

Ознака паритету P напряду залежить від поточного значення акумулятора. Встановлюється і скидається в кожному циклі команди і фіксує непарне/парне число одиничних бітів в акумуляторі. Якщо число одиничних бітів акумулятора непарне, то ознака P встановлюється, а якщо парне – скидається.

АЛП не керує ознаками селекції банку регістрів (RS0, RS1), і їх значення повністю визначається програмою та використовуються для вибору одного із чотирьох регістрових банків.

Ознака F0. Встановлюється та скидається програмно, стан біту може бути перевірений програмно. Використовується як ознака користувача.

3.4. Пам'ять програм в MCS-51

Пам'ять розміщена на кристалі носить назву **резидентної**. В MCS-51 резидентна пам'ять програм і пам'ять даних фізично і логічно розділені, мають різні механізми адресації, працюють під керуванням різних сигналів і виконують різні функції.

Резидентна пам'ять програм (РПП) має ємність 4 Кб і призначена для зберігання команд і констант.

РПП має 16-бітну шину адреси, через яку забезпечується доступ з лічильника команд або з регістра-вказівника даних. Останній виконує функцію базового регістра при непрямих переходах по програмі або використовується в командах, що оперують з таблицями. Початкові адреси пам'яті програм призначені для реакції на скидання та переривання.

У пам'яті програм мікроЕОМ зарезервована група адрес, на котрі здійснюється перехід при виникненні скидання чи переривання (табл. 33).

Табл. 3. Зарезервовані адреси у пам'яті програм

Адреса	Умова, за якою здійснюється перехід	Адреса	Умова, за якою здійснюється перехід
0000H	сигнал скидання (RST)	0013H	сигнал переривання INT1
0003H	сигнал переривання INT0	001BH	переповненням таймера 1
000BH	переповненням таймера 0	0023H	переривання від послідовного порту

3.5. Пам'ять даних в MCS-51

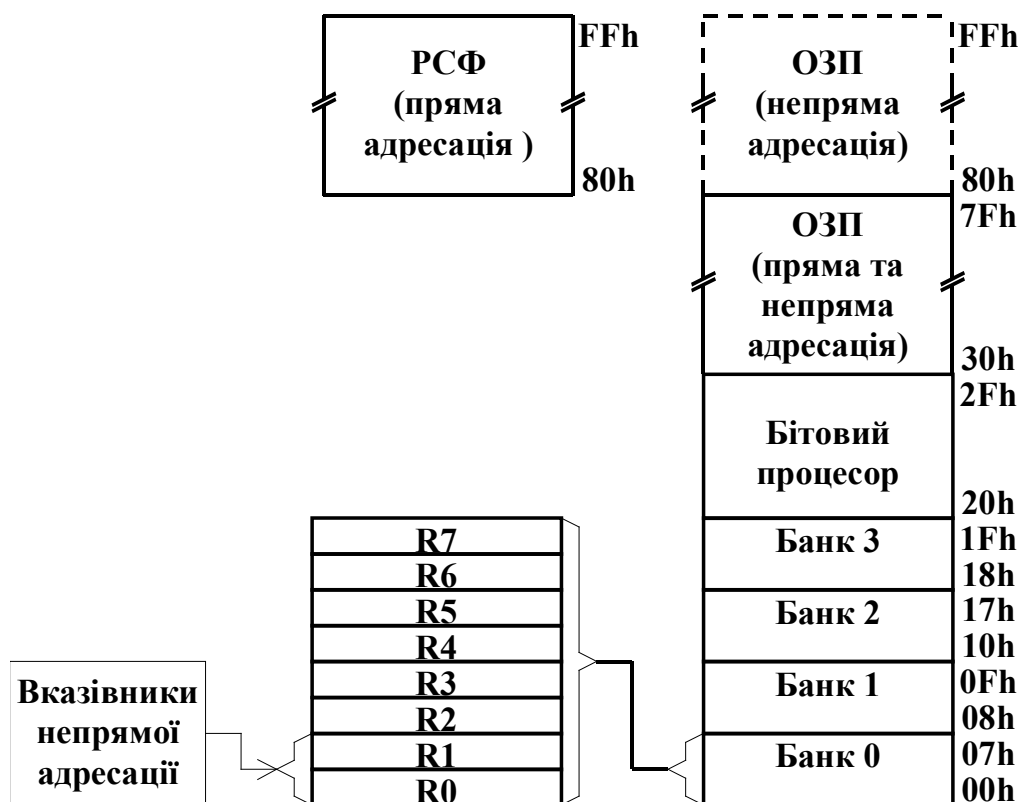


Рис. 3. Структура внутрішньої пам'яті даних

Резидентна пам'ять даних (РПД) представляє собою ОЗП. Структура внутрішньої пам'яті даних та регістрів, відображених на ній, зображена на рис.3. На внутрішню пам'ять даних відображаються два функціонально розділених блоки – оперативний запам'ятовуючий пристрій (ОЗП) ємністю 128÷256 байтів, призначений для збереження даних, і блок регістрів спеціальних функцій РСФ, що містить набір регістрів, які встановлюють режими роботи деяких вузлів ОМЕОМ. Адреси регістрів спеціальних функцій (РСФ) перелічені в таблиці 4.

Пам'ять програм, як і пам'ять даних, може бути розширена до 64 Кб шляхом підключення зовнішніх ВІС.

Основне ОЗП ОМЕОМ i80C51 займає адресний простір 0...7Fh окремі комірки допускають як пряму (за номером комірки), так і непряму адресацію. Регістри спеціальних функцій адресуються у просторі 80h...0FFh, але допускають лише пряму адресацію.

3.5.1. Регістри загального призначення MCS-51

Перші 32 байти ОЗП утворюють 4 банки регістрів. Один з банків може бути зроблений активним за допомогою встановлення відповідної комбінації бітів S1S0 у регістрі слова стану. У активному банку регістри допускають адресацію за назвами R0-R7. Регістри розміщуються у пам'яті у порядку збільшення їх номерів. Регістри R0 і R1 у кожному банку можуть бути використані як джерела адреси при непрямій адресації.

Наприклад, якщо в комірку пам'яті з номером 9 потрібно записати число 37, то при використанні прямої адресації така дія виконується командою:

MOV A,#9

де MOV – код операції пересилання; A – назва регістра, що є адресою призначення; #9 – число 9, символ # покликаний відрізнити число від номера комірки пам'яті.

На відміну від розглянутої команди, виконання команди

MOV A, 9

скопює у акумулятор вміст комірки пам'яті з адресою 9. Команда виконується процесором за 2 машинні цикли.

3.5.2. Область бітового процесора

На відміну від багатьох інших мікроЕОМ мікросхеми сімейства MCS-51 допускають маніпуляції із окремими бітами даних, що розміщені у спеціальній області пам'яті даних - так званому бітовому процесорі.

Бітовий процесор - це область пам'яті довжиною 32 байтів, що дозволяє звертання до окремих бітів. Молодші 16 байтів бітового процесора розташовані у послідовно розміщених 16 комірках пам'яті починаючи із адреси 20H. Розміщені у цих комірках пам'яті бітові змінні можуть бути використані у якості масиву бітових змінних, що відображають стан мікроконтроллера, системи, чи програми.

3.6. Регістри спеціальних функцій

До складу блока регістрів спеціальних функцій (PCF) входять:

- побітно адресовані акумулятор ACC (адреса 0E0H) і регістр В (адреса 0F0H);
- побітно адресовані регістри портів: P0 (адреса 80H), P1 (адреса 90H), P2 (адреса 0A0H) і P3 (адреса 0B0H);
- побітно адресований регістр слова стану PSW (адреса 0D0H);
- побітно адресовані регістри керування дозволом IE (адреса 0A8H) і пріоритетом IP (адреса 0B8H) переривань;
- регістри таймерів/лічильників TH1 (адреса 8DH), TH0 (8CH), TL1 (8BH), TL0 (8AH), регістр режимів TMOD (89H) і побітно адресований регістр керування роботою TCON (88H);
- показчик даних DPTR, представлений парою регістрів - старшим DPH (адреса 83H) і молодшим DPL (адреса 82H);
- показчик стеку SP (адреса 81H);
- регістр обміну SBUF (адреса 99H) і побітно адресований регістр керування SCON (адреса 98H) послідовного порту;
- регістр керування споживанням потужності PCON.

Табл. 4. Блок регістрів спеціальних функцій

Символ	Найменування	Адреса	Символ	Найменування	Адреса
1	2	3	1	2	3
ACC*	Акумулятор	0E0H	IE *	Регістр маски переривань	0A8H
B*	Регістр-розширювач акумулятора	0F0H	TMOD	Регістр режиму таймера/лічильника	89H
PSW*	Слово стану програми	0D0H	TCON*	Регістр керування/статус таймера	88H
SP	Регістр - вказівник стеку	81H	TH0	Таймер 0 (старший байт)	8CH
DPH	Старший байт вказівника даних (DPH)	83H	TL0	Таймер 0 (молодший байт)	8AH
DPL	Молодший байт вказівника даних (DPL)	82H	TH1	Таймер 1 (старший байт)	8DH
P0*	Порт 0	80H	TL1	Таймер 1 (молодший байт)	8BH
P1*	Порт 1	90H	SCON*	Регістр керування прийомо-передавачем	98H
P2*	Порт 2	0A0H	SBUF	Буфер прийомопередавача	99H
P3*	Порт 3	0B0H	PCON	Регістр керування потужністю	87H
IP*	Регістр пріоритетів	0B8H	* Регістри, імена яких відмічені знаком (*), допускають адресацію окремих бітів		

Звертання до регістрів РСФ може бути здійснене за їх абсолютними адресами, однак значно зручніше використовувати для адресації символічні власні імена регістрів РСФ, наведені вище.

При звертанні до РСФ за символічними іменами варто мати на увазі одну особливість адресації акумулятора.

Якщо акумулятор адресується як джерело або приймач байту, то для адресації використовується символічне ім'я А. Якщо ж адресуються окремі біти акумулятора, то в якості його імені використовується символ АСС.

Наприклад, записування числа 18 в акумулятор може бути здійснено командою

MOV A, #18, а встановлення в одиницю біта 7 – командою SETB ACC.7.

3.7. Регістри вказівники

В архітектурі MCS-51 є наступні регістри вказівники:

- 8-бітний вказівник стеку (SP)
- 16 бітний регістр - вказівник даних (DPTR).

8-бітний вказівник стеку SP може адресувати будь-яку область РПД. Його вміст інкрементується, перш ніж дані запам'ятовуються в стеку в ході виконання команд PUSH і CALL.

Вміст SP декрементується після виконання команд POP і RET. В процесі ініціалізації MCS51 після сигналу RST в SP автоматично завантажується код 07H. Це значить, що якщо прикладна програма не перевизначить стек, то перший елемент даних в стеку буде розміщуватись в комірці SP з адресою 08H.

Двобайтний регістр - вказівник даних (DPTR) звичайно використовується для фіксації 16-бітної адреси в операціях із зверненням до зовнішньої пам'яті. Командами MCS51 регістр-вказівник даних може бути використаний або як 16-бітний регістр, або як два незалежних 8-бітних регістри (DPH і DPL).

3.8. Порти вводу - виводу

Чотири порти MCS51 призначені для вводу або виводу інформації в байтовому форматі. Вихідні лінії портів P0 і P2, а також вхідний буфер порту 0, використовується при звертанні до зовнішньої пам'яті програм (ЗПП та даних (ЗПД)). При цьому через порт 0 спочатку виводиться молодший байт адреси, а потім видається або приймається байт даних.

Порт P0 використовується у якості шини адреси та даних, або у якості звичайного порту вводу-виводу.

У режимі шини дані не фіксуються у будь-якому внутрішньому регістрі, а кожна лінія порту P0 має високу та приблизно однакову навантажувальну здатність при видачі лог. 0 та лог. 1. В режимі звичайного порту дані, що виводяться, фіксуються у внутрішньому регістрі порту, а сам порт перетворюється у порт з відкритим колектором, котрий працює без внутрішніх резисторів. Якщо необхідно видати на лінію високий логічний рівень слід використати зовнішній підтягуючий резистор.

Через порт P2 видається старший байт адреси комірки пам'яті чи порту вводу-виводу при використанні 16-бітних адрес. Адреса ніяк не фіксується у схемі порту і не може бути зчитана. Після закінчення циклу обміну стан порту відновлюється з даних, записаних у внутрішньому регістрі.

У портах P1, P2, P3 високий рівень на лініях порта при виконанні основної функції формується підтягучим резистором паралельно до котрого під'єднаний транзистор, що не показаний на схемі. Транзистор вмикається лише на короткий час у момент зміни сигналу на лінії загострюючи додатній фронт сигналу.

Кожна лінія порту P3 має альтернативне призначення. У випадку використання альтернативної функції порту P3 стан лінії, створений альтернативною функцією, ніяк не фіксується, після видавання лог. 0 альтернативною функцією порт повертається до видавання даних записаних у порт. Альтернативне призначення можна використовувати при довільному стані лінії, але щоб воно працювало вірно на відповідний біт порта має бути видана лог. 1. Альтернативні функції ліній P3.7 - P3.0 можуть бути задіяні при наявності запису 1 в відповідні біти регістра-засувки порту P3, в іншому випадку на виході буде постійно лог. 0, що не дозволить використати альтернативні функції порту.

Табл. 5. Альтернативні функції порту P3

Символ	Позиція	Ім'я та призначення
\overline{RD}	P3.7	Читання. Активний сигнал низького рівня формується апаратно при звертанні до зовнішніх портів даних
\overline{WR}	P3.6	Запис. Активний сигнал низького рівня формується апаратно при звертанні до зовнішніх портів даних
T1	P3.5	Вхід таймера/лічильника 1 або тест-вхід
T0	P3.4	Вхід таймера/лічильника 0 або тест-вхід
$\overline{INT1}$	P3.3	Вхід запиту переривання 1. Сприймається сигнал низького рівня або перехід з високого рівня в низький.
$\overline{INT0}$	P3.2	Вхід запиту переривання 0. Сприймається сигнал низького рівня або перехід з високого рівня в низький.
TXD	P3.1	Вихід передавача послідовного порту в режимі UART. Вихід синхронізації в режимі зсувного регістра
RXD	P3.0	Вхід приймача послідовного порту в режимі UART. Ввід/вивід даних в режимі зсувного регістра

Кожна лінія портів може бути використана незалежно для вводу або виводу інформації. По сигналу RST в регістрі - засувці всіх портів автоматично записуються одиниці, що настраюють їх тим самим на режим вводу.

Слід зауважити, що порти P0 і P2 не можуть бути використані для керування зовнішніми схемами або передачі інформації в випадку, якщо МК система має зовнішню пам'ять, зв'язок з якою організується через загальну шину адреси/даних, яка працює в режимі часового мультиплексування.

4. Доступ до зовнішньої пам'яті

В системах, побудованих на основі MCS51, можливе використання двох типів зовнішньої пам'яті: зовнішньої пам'яті програм (ЗПП) та зовнішньої пам'яті даних (ЗПД).

Доступ до ЗПП здійснюється за допомогою керуючого сигналу - PSEN, що виконує функцію сигналу читання.

Доступ до ЗПД забезпечується керуючими сигналами WR та RD, які формуються в лініях P3.7 и P3.6 при виконанні портом P3 альтернативних функцій (**Ошибка! Источник ссылки не найден.**).

При звертанні до ЗПП завжди використовується 16-бітна адреса. Доступ до ЗПД можливий з використанням 16-бітної адреси (MOVX A, @DPTR) або 8-бітної адреси (MOVX A, @Ri).

В будь-якому випадку, при використанні 16-бітної адреси старший байт адреси фіксується (та зберігається незмінним на протязі одного циклу запису або читання) в регістрі-засувці порту P2.

5. Програмування однокристаліної мікро-ЕОМ МК51 з використанням програмного емулятора Single-Chip Machine

Система моделювання Single-Chip Machine є програмним емулятором для мікроконтролера MCS51 і призначена для:

- моделювання роботи мікро-ЕОМ MCS51 в сукупності з мікросхемою-розширником портів вводу-виводу KP580BP43 і блоком зовнішньої пам'яті даних об'ємом 256 байт;
- розробки та налагодження програм для мікроконтролерів серії МК51;
- дослідження поведінки внутрішніх і зовнішніх сигналів зазначених мікросхем.

Інтерфейс емулятора зображено на рис 4.

Для завантаження емулятора потрібно з каталогу дистрибутива запустити файл setup.exe. Далі необхідно ознайомитися і слідувати вказівкам майстра установки.

Для запуску програми необхідно натиснути кнопку «Пуск», після чого в меню «Програми» вибрати папку з заданим при інсталяції ім'ям (SCM за замовчуванням).

У цій папці знаходяться два ярлики, а саме:

- SCM Help для виклику довідкової системи і
- Single-Chip Machine 1.xx для виконуваного модуля.

Для запуску програми необхідно вибрати ярлик Single-Chip Machine 1.xx і натиснути на ньому лівою кнопкою мишки.

SCM включає засоби налагодження і редагування програм на асемблері з вбудованим інтерпретатором, що робить введення програм набагато зручнішим і ефективнішим, ніж в інших емуляторах подібного класу.

Виконання програми користувача здійснюється з максимальним наближенням до дійсності за допомогою імітаційної моделі, рівень деталізації якої дорівнює одному такту ($1t = 0.5 \text{ mks}$).

Можна налаштувати режим моделювання:

- на один такт вперед;
- на один машинний цикл вперед;
- на один крок вперед;
- виконання кроку до зміни регістра адреси мікроконтролера;
- виконання до найближчої точки зупину;
- виконання до кінця програми;
- виконання до першої порожньої комірки пам'яті;
- на один машинний цикл назад;
- на один такт назад.

Крім того, користувачеві надаються такі можливості як:

- часові діаграми внутрішніх і зовнішніх сигналів;
- імітація зовнішніх сигналів з відображенням змін на умовно-графічному відображенні мікросхем;

- можливість зміни значень вузлів мікроЕОМ в процесі роботи моделі та ін.

Вбудований редактор-компілятор дозволяє набирати програми на асемблері МК51, а потім за допомогою кнопки «компіляція» перевести текст програм в машинні коди і записати його, як в файл ПЗП з розширенням ".MPM", так і в ПЗП мікроконтролера для відображення в відлагоджувачі.

Крім формату ПЗП, існує ще один формат пам'яті програм - формат «hex», який підтримується програмним забезпеченням всіх моделей програматорів.

Таким чином, SCM повністю сумісний з промисловими емуляторами.

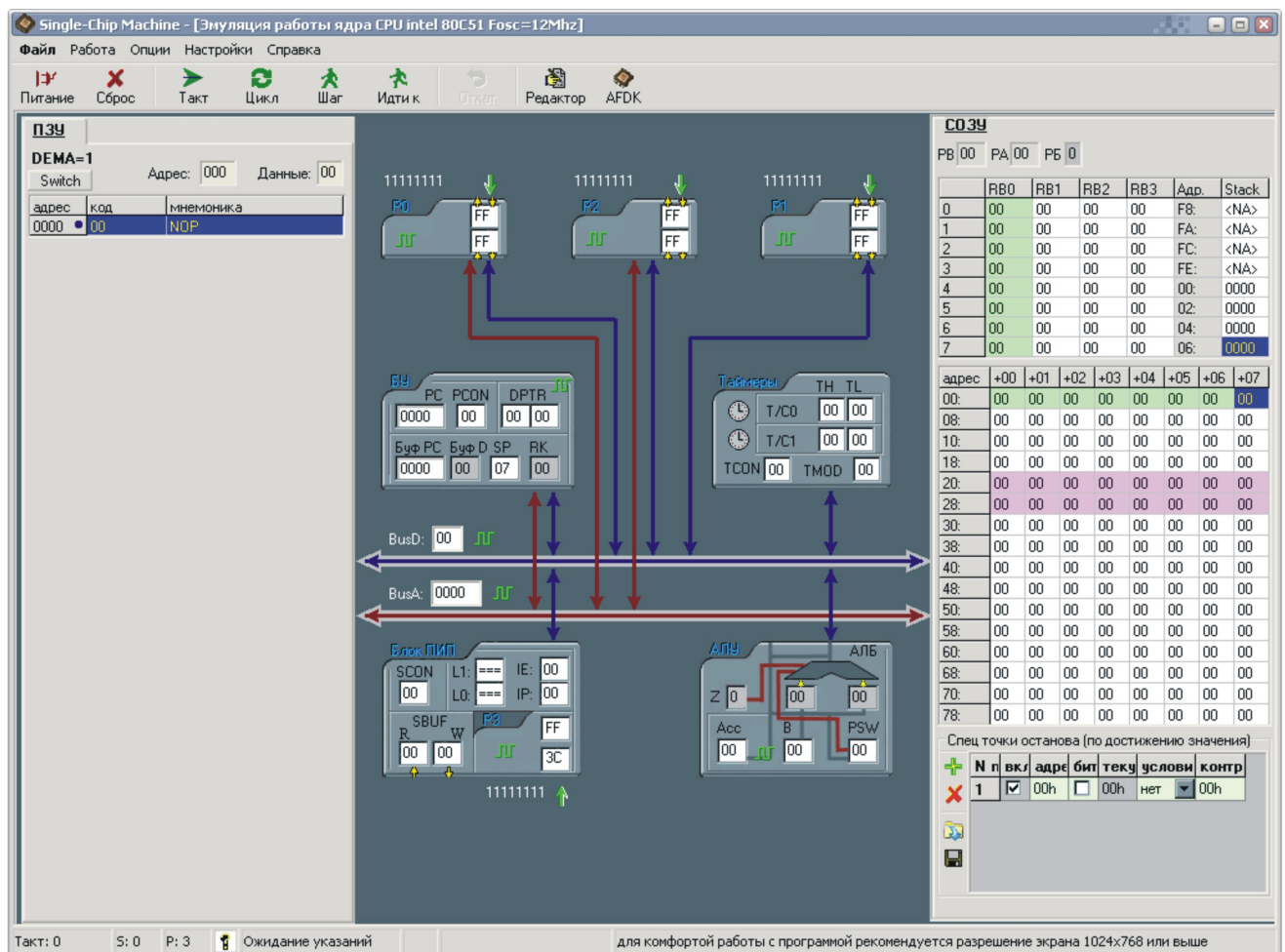


Рис. 4

Налаштування емулятора можна змінити двома шляхами.

Перший - за допомогою меню «Налаштування» і вибором відповідного пункту.

Другий - за допомогою самостійного редагування файлу SCMF.CFG, текстового файлу конфігурації програми SCN.

Якщо програму завантажити неможливо, то буде видано відповідне повідомлення, в якому пояснюється причину.

Інтерфейс (вікно) емулятора містить (рис. 4):

у центральній частині:

- зображення трьох портів P0, P1, P2, внизу зліва – P3 (молодший біт портів – справа);
- програмний лічильник PC;
- регістр керування потужністю контролера PCON;
- показчик даних DPTR;
- буфер адресної шини БуфРС;
- буфер шини даних БуфD;
- вказівник стеку SP;
- регістр команд RK;
- два таймери-лічильники T/C0 та T/C1;
- регістр керування таймером-лічильником TCON;
- регістр режиму таймера-лічильника TMOD;
- регістр керування прийомо-передавачем SCON;
- регістр маски переривання IE;
- акумулятор ACC;
- ознака нульового значення вмісту акумулятора Z;
- регістр-розширювач акумулятора B;
- буфер прийомопередавача SBUF;
- тригер заборони переривання IE;
- флажок користувача F1;

- регістр слова-стану програми PSW: C – флажок перенесення, AC – флажок допоміжного перенесення, F0 – флажок користувача, RS0, RS1 – селектор банків регістрів, 0V – ознака переповнення. P – ознака паритету.

В лівій верхній частині інтерфейсу висвічується адреси та дані внутрішньої пам'яті програм (РПЗП).

В правій верхній частині розташована резидентна пам'ять даних (РПД), початкову область якої займають регістри R0...R7 чотирьох банків RB0... RB3, а також стек.

Під ними розташовується резидентна ОЗП.

Для завантаження вже існуючого файлу необхідно натиснути курсором на верхню ліву закладку «Файл», і далі – «Відкрити». З папки «Source» вибрати для завантаження файл на асемблері, з папки «Work» - об'єктний файл (з розширенням hex).

Для створення нового файлу потрібно курсором натиснути на опцію «Редактор» (знаходить в верхній частині інтерфейсу на панелі управління). З'явиться поле редактора (рис. 5), в якому записуються команди **асемблера** Вашої робочої програми.

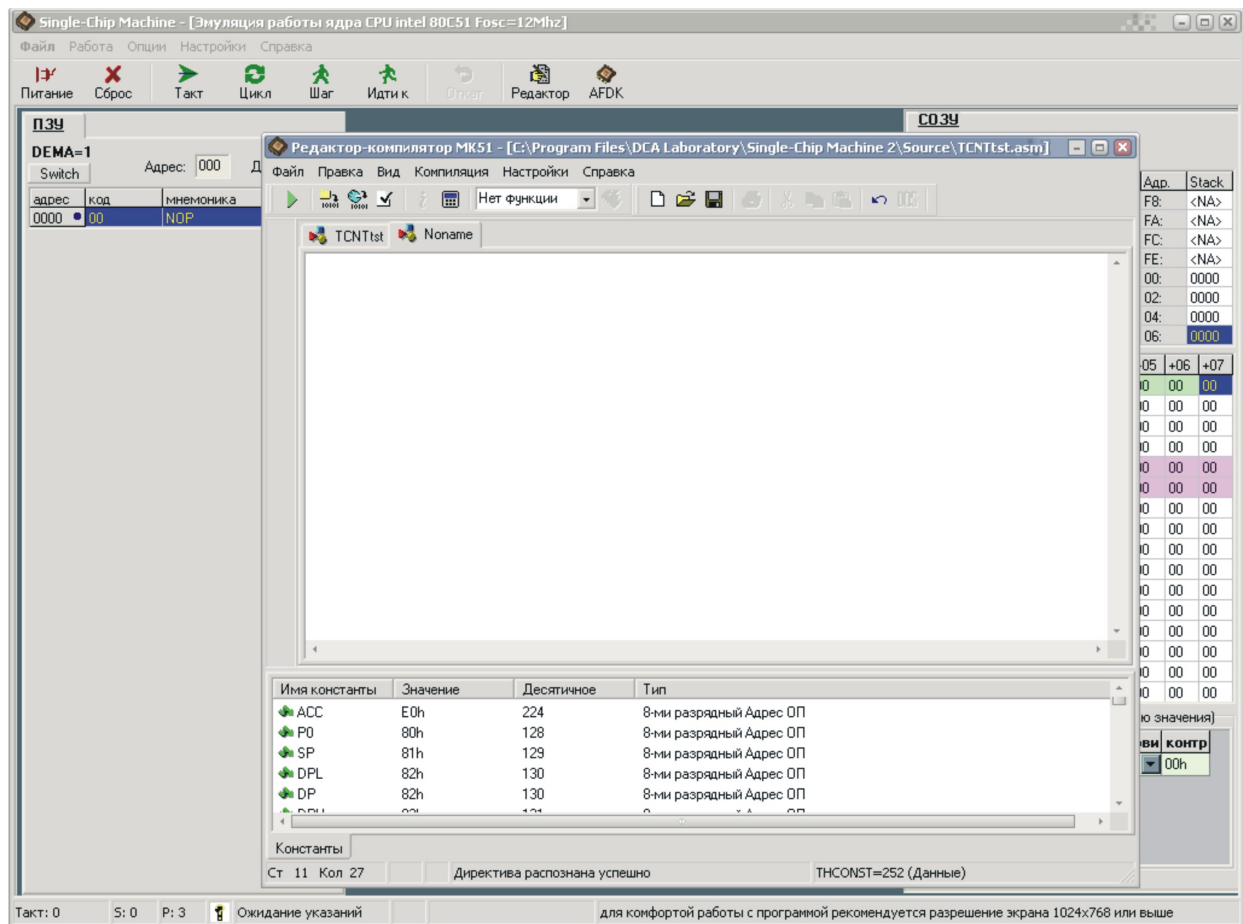


Рис. 5

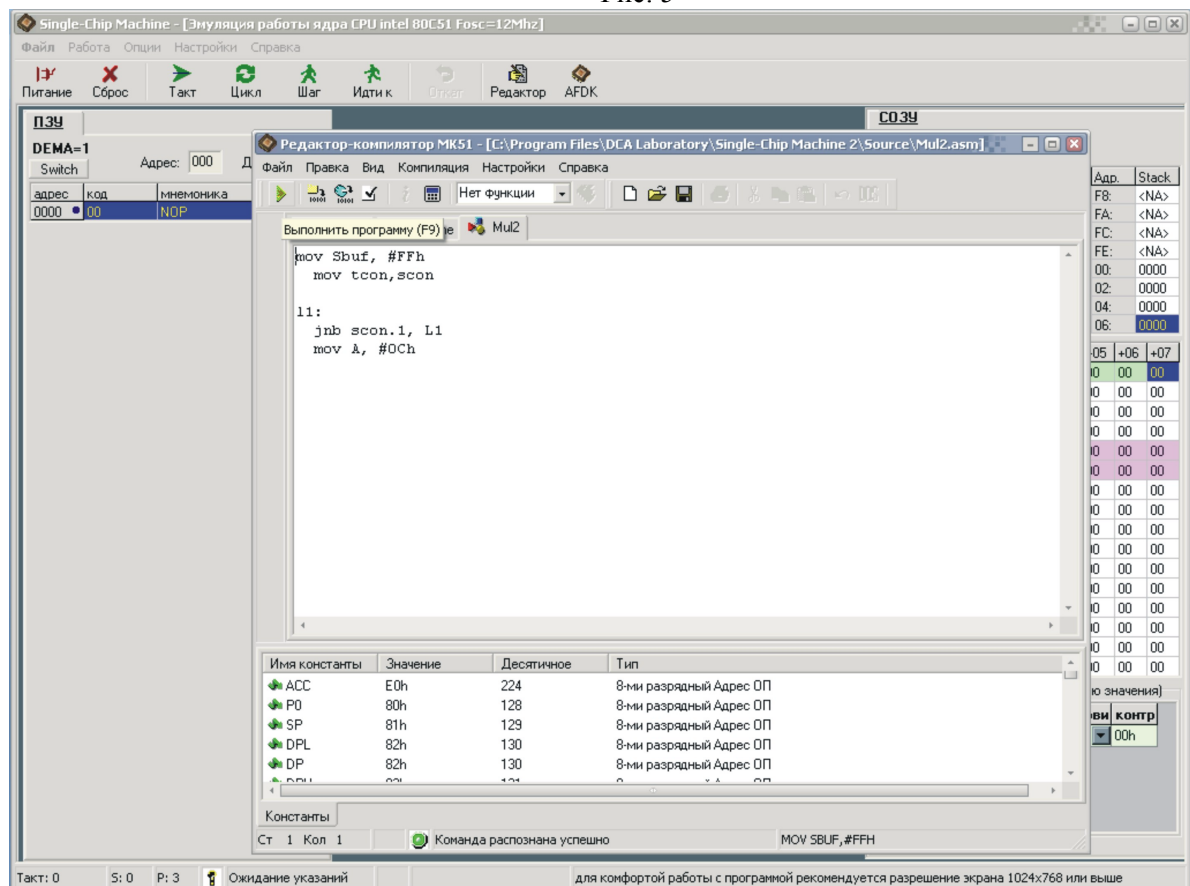


Рис. 6

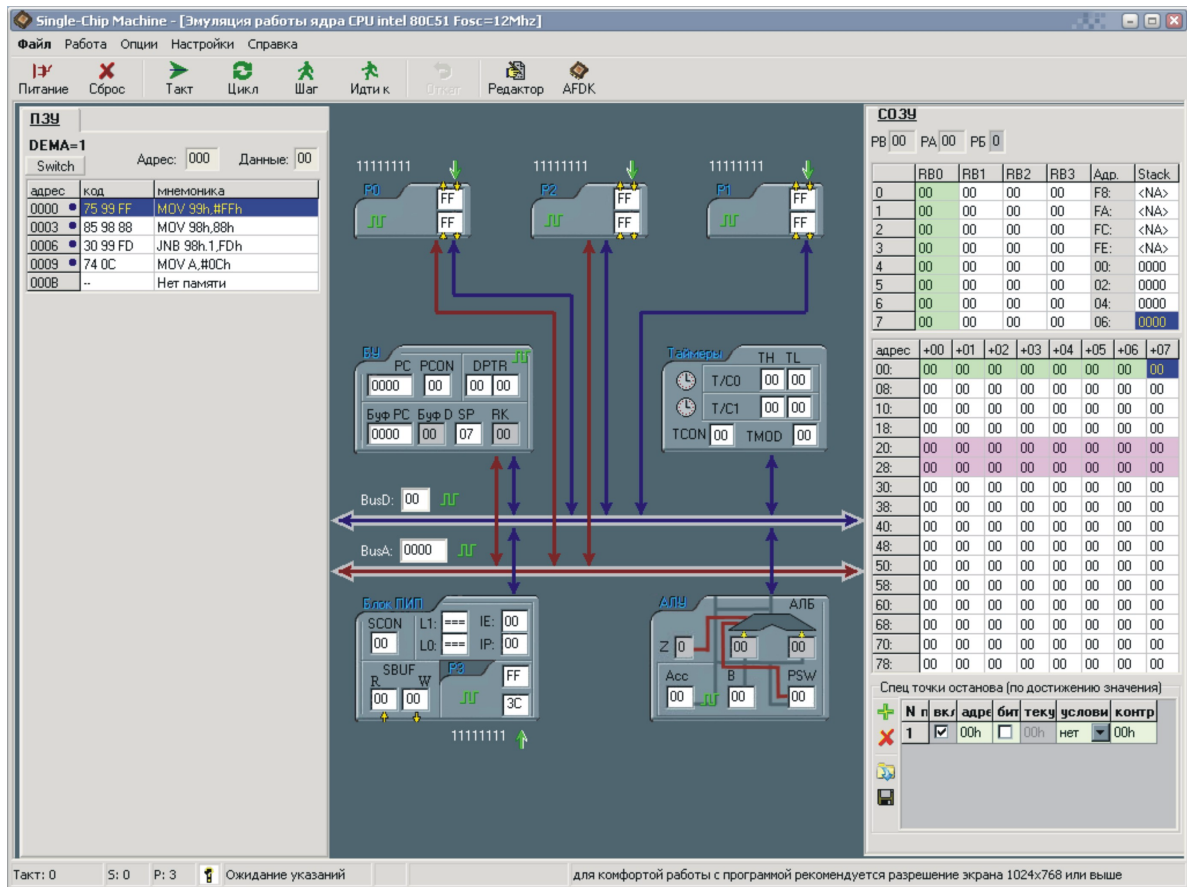


Рис. 7

Для запуска программы на выполнение необходимо нажать на изображении трикутника зеленого колыору («Запуск редактора») в верхней части окна редактора (рис. 6).

Для компиляции программы необходимо нажать курсором на кнопку справа от трикутника («Компиляция»), после чего скомпилированная программа появится на левом поле эмулятора (поле ПЗП) (рис. 7).

Для запуска программы на выполнение можно воспользоваться закладкой «Крок» на панели управления эмулятора (кроковый режим) либо закладкой «Йти до» (программа выполнится до заданной команды).

Вернуться на начало программы можно закладкой «Скидання».

Приклад программы для работы с эмулятором контролера

Sjmp Start

Org 0Bh

RetI

Org 1bh

RetI

Org 23h

Start:

Mov IE, #FFh

Mov IP, #Ch

Mov Th0, #E0h

Mov Tl0, #E0h

mov TMod, #33h

; Таймер 0 переводимо в режим 3

mov TCon, #50h

; Запуск обоих каналов таймера TCNT0

L2:

nop

```

jnb tf0,L2
L3:      Djnz R1,L3          ;sjmp L3

      Nop
      nop
      nop
      nop
End

```

5. 1. ЗАВДАННЯ для виконання лабораторної роботи

Завдання 1. Записати в резидентну пам'ять даних за адресами 41 і 42 число 1С3FH:

```

LOAD:
      MOV R0,#41H          ;завантаження в R0 покажчика даних
      MOV @R0,#1CH         ;завантаження в пам'ять числа 1CH
      INC R0               ;інкремент покажчика
      MOV @R0,#3FH         ;записати в пам'ять число 3FH

START:
      LJMP LOAD1           ;перехід до програми LOAD
      ORG 2100H            ;директива розташування програми за адресою 2100

LOAD1:
      MOV R0,#41H          ;завантаження в R0 покажчика даних
      MOV @R0,#1CH         ;записати в пам'ять число 1CH
      INC R0               ;інкремент покажчика
      MOV @R0,#3FH         ;записати в пам'ять число 3FH
      LJMP LOAD            ;за циклювання програми
      END                  ;директива закінчення трансляції

```

Завдання 2. Додавання. Додати два двійкових багатобайтних числа. Доданки розташовуються в резидентній пам'яті даних, починаючи з молодшого байту. Початкові адреси доданків задані в R0 і R1, формат доданків в байтах - в R2:

```

      CLR C                ;скидання ознаки перенесення
LOOP:  MOV A,@R0            ;завантаження в A поточного байту першого доданка
      ADDC A,@R1            ;додавання байтів з врахуванням перенесення
      MOV @R0,A            ;розміщення байту результату
      INC R0               ;просування покажчиків
      INC R1
      DJNZ R2, LOOP        ;цикл, якщо не всі байти просумовані

```

Завдання 3. Потрібно помножити ціле двійкове число довільного формату на константу 73. Початкове число розташовується в резидентній пам'яті даних, адреса молодшого байту знаходиться в регістрі R0. Формат числа в байтах зберігається в R1:

```

      MOV A,#0             ;скидання акумулятора
LOOP:  ADD A,@R0            ;завантаження множеного
      MOV B,#73            ;завантаження множника
      MUL AB               ;множення
      MOV @R0,A            ;запис молодшого байту часткового добутку
      INC R0               ;інкремент адреси
      MOV A,B              ;пересилання старшого байту часткового
                          ;добутку в акумулятор
      XCH A,@R0            ;попереднє формування наступного байту добутку
      DJNZ R1,LOOP        ;цикл, якщо не всі байти початкового числа помножені
                          ;на константу

```

Завдання 4. Операції із стеком

```

START:
      MOV R1,#02H          ;завантаження регістрів
      MOV A,#30H
      MOV R2,#00H

```

```

        LCALL SUB          ;перехід на підпрограму
        SJMP START

SUB:
        PUSH PSW           ;зберігання в стеку вмісту PSW
        PUSH ACC           ;зберігання акумулятора
        PUSH B             ;зберігання вмісту B в акумуляторі
        ADD A,R1           ;обробка даних
        MOV R2,A
        POP B              ;відновлення B
        POP ACC            ;відновлення акумулятора
        POP PSW            ;відновлення PSW
        RET                ;повернення
        END

```

Завдання 5. Використання логічних команд

```

OUT:
        MOV P1,#10101010B ;вивід байту в порт P1
        SETB P1.0          ;встановлення в «1» нульового розряду
        CLR P1.3           ;скидання в «0» 3-го розряду
        CLR P1.4           ;скидання в «0» 4-го розряду
        CLR P1.5           ;скидання в «0» 5-го розряду
        CLR P1.6           ;скидання в «0» 6-го розряду
        CLR P1.7           ;скидання в «0» 7-го розряду
        XRL P1, #11110000B ;інверсія розрядів 4 – 7
        ORL P1, #00001000B ;встановлення в «1» третього розряду
        ANL P1, #11111110B ;скидання в «0» нульового розряду

```

Для кожного з варіантів завдання після виконання кожної з команд записати вміст тих регістрів, який змінюються в процесі виконання програми, в табл.1.

Таблиця 1

Регістр	PC	P0	P1	P2	A	B	PSW	C	AC	SP
Команда 1														
Команда 2														
.....														
Команда n														

6. Контрольні запитання

1. Основні технічні характеристика МК i8051.
2. Призначення регістрів A та B.
3. Призначення регістрів PC та DPTR.
4. Особливості будови та використання портів P0, P1, P2 та P3.
5. Банки регістрів загального призначення: будова та застосування.
6. Пояснити роботу команд MOV, MOVC та MOVX.
7. Назвати та пояснити роботу арифметичних команд.
8. Назвати та пояснити роботу логічних команд.
9. Назвати та пояснити роботу команд безумовних та умовних переходів.

7. Література

1. Горюнов А.Г. Ливенцов С.Н. Система команд микропроцессора: Томский политехнический университет. Учеб. пособие. Томск – 2004.
2. Проектирование цифровых устройств на однокристальных микроконтроллерах / В. В. Сташин и др. - М.: Энергоатомиздат, 1990. – 224 с.
3. Микропроцессоры / Под ред. Преснухина Л.Н., т. 1, 2, 3. - М.: Высшая школа, 1986.
4. Бойко Н.П., Стеклов В.К. Системы автоматического управления на базе микро-ЭВМ. - К.: Техника, 1989. - 182 с.

Додаток 1

Система команд ОМЕОМ МК51

Мікроконтролери сімейства МК51 підтримують систему з 111 базових команд, які можна розбити на групи, розглянуті нижче.

Надалі використовуються наступні позначення:

Rn, n=0...7- реєстр поточного реєстрового банку;

Ri, i=0.1 - реєстри R0 чи R1 поточного реєстрового банку, використовувані при непрямій адресації;

#d - число довжиною 1 байт (8біт);

#d16 - число довжиною 2 байти;

ad - пряма адреса в ОЗП чи даних;

bit - адреса біту (у бітових операціях);

PC - лічильник команд;

(ad) - вміст комірки пам'яті з адресою (ad);

((Ri)) - вміст комірки пам'яті, адреса якого знаходиться в Ri;

<- стрілка вказує розміщення результату операції.

При виконанні арифметичних операцій операнди і результати операції подаються в стандартних для восьмирозрядних обчислювальних систем форматах, причому молодший розряд числа відповідає розряду D0, а старший - D7.

1. Група команд передачі даних

Пересилання в межах резидентного зупд здійснюються командою **MOV X,Y**, де X - адреса призначення, за якою записується число чи операнд, зазначений як Y. Вміст комірки пам'яті, зазначеної як операнд Y, при виконанні операції MOV не змінюється.

Об'єм пам'яті в ЗУПП, що займається командами MOV, швидкість їхнього виконання в машинних циклах, а також виконувані функції приведені в таблиці 1.

Таблиця 1

Мнемокод	Б	Ц	Операція
MOV A, Rn	1	1	(A)<(Rn)
MOV A, ad	2	1	(A)<(ad)
MOV A, @Ri	1	1	(A) <((Ri))
MOV A, #d	2	1	(A) < #d
MOV Rn, A	1	1	(Rn) < (A)
MOV Rn, ad	2	2	(Rn) < (ad)
MOV Rn, #d	2	1	(Rn) < #d
MOV ad, A	2	1	(ad) < (A)
MOV ad, Rn	2	2	(ad) < (Rn)
MOV add, ads	3	2	(add) < (ads)
MOV ad, @Ri	2	2	(ad) < ((Ri))
MOV ad, #d	3	2	(ad) < #d
MOV @Ri, A	1	1	((Ri)) < (A)
MOV @Ri, ad	2	2	((Ri)) < (ad)
MOV @Ri, #d	2	1	((Ri)) < #d
MOV DPTR, #d16	3	2	(DPTR) < #d16

Пересилання за межами резидентної пам'яті програм здійснюються командою MOVC (MOVE Code memory), а з/у зовнішню пам'ять даних – командою MOVX (MOVE eXternal memory).

Система підтримуваних процесором команд вигляду MOVX і MOVC зведена в таблицю 2.

Таблиця 2

Мнемокод	Операція
MOV A, @A+DPTR	$(A) < ((A) + (DPTR))$
MOVC A, @A+PC	$(PC) < (PC) + 1; (A) < ((A) + (PC))$
MOVX A, Ri	$(A) < ((Ri))$
MOVX A, @DPTR	$(A) < ((DPTR))$
MOVX @Ri, A	$((Ri)) < (A)$
MOVX @DPTR, A	$((DPTR)) < (A)$

2. Команди обміну мають мнемо код XCH A, X

При виконанні цієї команди вміст акумулятора міняється місцем із вмістом комірки пам'яті, зазначеної як операнд X. Допустимі види команд обміну і їх характеристики зведені в таблицю 3.

Таблиця 3

Мнемокод	Б	Ц	Операція
XCH A, Rn	1	1	$(A) \leftrightarrow (Rn)$
XCH A, ad	2	1	$(A) \leftrightarrow (ad)$
XCH A, @Ri	1	1	$(A) \leftrightarrow ((Ri))$

Крім зазначених команд, до групи команд передачі даних прийнято відносити операції

PUSH ad - запис в стек і

POP ad - витяг зі стеку.

Ці команди виконуються за 2 машинних цикли і займають у пам'яті програм по 2 байти.

Стек - це область ОЗП зі спрощеним доступом за адресою, автоматично формованою вказівником стеку SP. Відмінність роботи стеку ОМЕОМ сімейства МК51 полягає в тому, що при записі інформації в стек спочатку інкрементується (збільшується на одиницю) вміст SP, а потім за адресою, зазначеною в ньому, записується операнд, зазначений у команді.

Витяг операнду зі стеку здійснюється в зворотньому порядку.

Варто мати на увазі, що після рестарту процесора в SP записується число 7, і, якщо в програмі використовуються переривання, чи підпрограми команди PUSH, для запобігання втрати інформації, записаної в регістри банку 1, варто змінити значення SP (положення вершини стеку).

3. Група команд арифметичних операцій містить у собі:

ADD A,X - додавання;

ADDC A,X - додавання з врахуванням перенесення;

SUBB A,X - віднімання з врахуванням запозичення;

INC X - одиничне позитивне збільшення (інкремент);

DEC X - одиничне негативне збільшення (декремент);

MUL AB - беззнакове множення;

DIV AB - беззнакове ділення;

DA A - десяткову корекцію акумулятора.

Команда **ADD** реалізує функцію обчислення суми вмісту акумулятора з операндом X (або як цілих чисел без знаку, або як цілих зі знаком у додатковому коді) і запис результату в акумулятор. Вміст операнду X (якщо це комірка пам'яті) при цьому не зміниться. При виконанні команди формується ознака перенесення C.

Команда **ADDC** (**SUBB**) на додаток до обчислення суми (різниці) враховує перенесення між байтами і може бути використана для обробки операндів, довжина яких перевищує 8 розрядів.

Команда **MUL AB** перемножує цілі числа без знаків з акумулятора і регістра B.

Добуток двох восьмирозрядних чисел буде мати довжину 16 розрядів і займати 2 байти. Старший байт добутку міститься в регістрі В, а молодший - в акумуляторі. Якщо результат добутку перевищує 255, то встановлюється флажок переповнення OV в регістрі ознак, в іншому випадку він скидається. У будь-якому випадку скидається флажок переносу.

Команда **DIV AB** здійснює беззнакове ділення восьмибітного цілого числа з акумулятора на восьмибітне число з регістра В. Частка від ділення заноситься в акумулятор, залишок - у регістр В.

Флажки перенесення C і переповнення OV скидаються.

Основні види арифметичних операцій і їх характеристики зведені в таблицю 4.

Таблиця 4

Мнемокод	Б	Ц	Операція
ADD A, Rn	1	1	$(A) < (A) + (Rn)$
ADD A, ad	2	1	$(A) < (A) + (ad)$
ADD A, @Ri	1	1	$(A) < (A) + ((Ri))$
ADD A, #d	2	1	$(A) < (A) + \#d$
ADDC A, Rn	1	1	$(A) < (A) + (Rn) + (C)$
ADDC A, ad	2	1	$(A) < (A) + (ad) + (C)$
ADDC A, @Ri	1	1	$(A) < (A) + ((Ri)) + (C)$
ADDC A, #d	2	1	$(A) < (A) + \#d + (C)$
DA A	1	1	Якщо $(A_{0-3}) > 9 \vee ((C) = 1)$, то $(A_{0-3}) < (A_{0-3}) + 6$, потім якщо $(A_{4-7}) > 9 \vee ((C) = 1)$, то $(A_{4-7}) < (A_{4-7}) + 6$
SUBB A, Rn	1	1	$(A) < (A) - (C) - (Rn)$
SUBB A, ad	2	1	$(A) < (A) - (C) - (ad)$
SUBB A, @Ri	1	1	$(A) < (A) - (C) - ((Ri))$
SUBB A, #d	2	1	$(A) < (A) - \#d$
INC A	1	1	$(A) < (A) + 1$
INC Rn	1	1	$(Rn) < (Rn) + 1$
INC ad	2	1	$(ad) < (ad) + 1$
INC @Ri	1	1	$((Ri)) < ((Ri)) + 1$
INC DPTR	1	2	$(DPTR) < (DPTR) + 1$
DEC A	1	1	$(A) < (A) - 1$
DEC Rn	1	1	$(Rn) < (Rn) - 1$
DEC ad	2	1	$(ad) < (ad) - 1$
DEC @Ri	1	1	$((Ri)) < ((Ri)) - 1$
MUL AB	1	4	$(B)(A) < (A) \times (B)$
DIV AB	1	4	$(A).(B) < (A)/(B)$

4. Група команд логічних операцій включає команди логічного АБО (ORL), І(ANL) і виключне АБО (XRL), які реалізують відповідні порозрядні операції над вмістом операндів.

Характерним для OMEOM сімейства МК51 є необов'язковість використання акумулятора в логічних командах в якості одного з джерел операндів, та як приймач результату операції.

Таблиця 5 дає узагальнення підтримуваних типів логічних команд і їх характеристики.

Мнемокод	Б	Ц	Операція
ANL A, Rn	1	1	$(A) \leftarrow (A) \wedge (Rn)$
ANL A, ad	2	1	$(A) \leftarrow (A) \wedge (ad)$
ANL A, @Ri	1	1	$(A) \leftarrow (A) \wedge ((Ri))$
ANL A, #d	2	1	$(A) \leftarrow (A) \wedge \#d$
ANL ad, A	2	1	$(ad) \leftarrow (ad) \wedge (A)$
ANL ad, #d	3	2	$(ad) \leftarrow (ad) \wedge \#d$
ORL A, Rn	1	1	$(A) \leftarrow (A) \vee (Rn)$
ORL A, ad	2	1	$(A) \leftarrow (A) \vee (ad)$
ORL A, @Ri	1	1	$(A) \leftarrow (A) \vee ((Ri))$
ORL A, #d	2	1	$(A) \leftarrow (A) \vee \#d$
ORL ad, A	2	1	$(ad) \leftarrow (ad) \vee (A)$
ORL ad, #d	3	2	$(ad) \leftarrow (ad) \vee \#d$
XRL A, Rn	1	1	$(A) \leftarrow (A) \oplus (Rn)$
XRL A, ad	2	1	$(A) \leftarrow (A) \oplus (ad)$
XRL A, @Ri	1	1	$(A) \leftarrow (A) \oplus ((Ri))$
XRL A, #d	2	1	$(A) \leftarrow (A) \oplus \#d$
XRL ad, A	2	1	$(ad) \leftarrow (ad) \oplus (A)$
XRL ad, #d	3	2	$(ad) \leftarrow (ad) \oplus \#d$

5. Група команд операцій з бітами

OMEOM сімейства МК51 підтримують групу операцій: скидання CLR, встановлення SETB, інверсію CPL, кон'юнкцію ANL, диз'юнкцію ORL і пересилання MOV операндів, в якості яких використовуються біти - окремі розряди побітноадресованих регістрів РСФ чи біти з області прямоадресованих бітів ОЗП.

Так, наприклад, за допомогою команди

SETB P1.0

можна встановити в стан логічної одиниці сигнал на виводі 1 OMEOM (рис. 1), а командою

MOV P2.5,C

вивести значення ознаки перенесення на вивід 26.

Для адресації бітів використовується пряма восьмирозрядна адреса; непряма адресація біт не підтримується.

Основні характеристики і допустимі комбінації бітових операцій зведені в таблицю 6.

Таблиця 6

Мнемокод	Б	Ц	Операція
CLR C			$(C) < 0$
CLR bit			$(bit) < 0$
SET B C			$(C) < 1$
SET B bit			$(bit) < 1$
CPL C			$(C) < (C)$
CPL bit			$(bit) < (bit)$
ANL C, bit			$(C) < (C) (bit)$
ANL C, /bit			$(C) < (C) (bit)$
ORL C, bit			$(C) < (C) \vee (bit)$
ORL C, /bit			$(C) < (C) \vee (bit)$
MOV C, bit			$(C) < (bit)$
MOV bit, C			$(bit) < (C)$

6. Група команд передачі керування включає стандартні для будь-якого процесора команди безумовного переходу, команди умовних переходів за значеннями ознак і біт, команди виклику / повернення з підпрограм, а також ряд оригінальних операцій, характерних тільки для OMEOM сімейства MK51.

Всі команди передачі керування виконуються за 2 машинних цикли.

Розрізняють 2 формати команди безумовних переходів і аналогічні їм 2 формати команди виклику підпрограм:

- LJMP <address> - безумовний перехід і
- LCALL <address> - виклик підпрограми в адресному просторі 64Кбайт.
- AJMP <address> - безумовний перехід і
- ACALL <address> - виклик підпрограми в адресному просторі 2Кбайт.

Команди AJMP і ACALL займають у порівнянні з LJMP і LCALL менше місця в ЗУПП - 2 байти замість трьох. Це обставина в деяких випадках може мати дуже важливе значення, оскільки будь-які умовні переходи можливі тільки на відстань + 127 байт у ЗУПП щодо коду команди умовного переходу.

Умовні переходи можуть виконуватися:

- за значенням біту перенесення C:
- JC <address> - переходити на <address>, якщо C=1, а
- JNC <address> - якщо C=0.
- за нульовим значенням акумулятора:

JZ <address> - переходити на <address>, якщо вміст акумулятора дорівнює 0 і

JNZ <address> - якщо не дорівнює 0.

- за значенням біту:

JB bit, <address> - переходити на <address>, якщо значення біта bit=1

JNB bit, <address> - переходити, якщо значення біта bit=0

JBC bit, <address> - переходити, якщо значення біта bit=1

і скидання значення цього біту в 0. Використання останньої з цих команд зручно при обслуговуванні переривань.

До команд умовних переходів відносять також команду організації циклу

DJNZ X, <address>

і команди чекання події

CJNE X, #d, <address>

CJNE A, ad, <address>

Команда DJNZ X,<address> віднімає одиницю з вмісту операнда X, у якості якого може бути зазначений регістр чи прямоадресована комірка пам'яті, зберігаючи його нове значення, і порівнює результат з нулем.

Команда CJNE X,#d,<address> порівнює значення операнда X, у якості якого можуть бути зазначені акумулятор, чи регістр з непрямою адресою комірки пам'яті, і числа #d, і, якщо вони не рівні, процесор переходить до виконання команди, розташованої за адресою <address>. Якщо ж рівні, то до виконання наступної за CJNE команди.

Дія команди CJNE A,ad,<address> аналогічна.

Повернення з підпрограми здійснюється за командою RET, розміщеної наприкінці підпрограми.

Команда RETI використовується для повернення процесора з підпрограми переривань; її відмінності від команди RET будуть розглянуті далі при описі процесу переривань.

Можливі конструкції команд передачі керування і їхня довжина приведені в таблиці 7, де:

ad 16 - шестнадцятирозрядна адреса (64 Кбайт);

ad 11 - одинадцятирозрядна адреса (2 Кбайт);

rel - восьмирозрядна адреса (127 байт);

Таблиця 7

Мнемокод	Б	Ц	Операція
LJMP ad 16	3		(PC) < ad 16

Мнемокод	Б	Ц	Операція
AJMP ad 11	2		$(PC) < (PC) + 2(PC_{0-10}) < ad\ 11$
SJMP rel	2		$(PC) < (PC) + 2(PC_{0-10}) < rel$
JMP @A+DPTR	1		$(PC) < (A) + (DPTR)$
JZ rel	2		$(PC) < (PC) + 2$, якщо $(A) = 0$, то $(PC) < (PC) + rel$
JNZ rel	2		$(PC) < (PC) + 2$, якщо $(A) \neq 0$, то $(PC) < (PC) + rel$
JC rel	2		$(PC) < (PC) + 2$, якщо $(C) = 0$, то $(PC) < (PC) + rel$
JNC rel	2		$(PC) < (PC) + 2$, якщо $(C) \neq 0$, то $(PC) < (PC) + rel$
JB bit, rel	3		$(PC) < (PC) + 3$, якщо $(bit) = 0$, то $(PC) < (PC) + rel$
JNB bit, rel	3		$(PC) < (PC) + 3$, якщо $(bit) \neq 0$, то $(PC) < (PC) + rel$
JBC bit, rel	3		$(PC) < (PC) + 3$, якщо $(bit) = 0$, то $(bit) < 0$ і $(PC) < (PC) + rel$
DJNZ Rn, rel	2		$(PC) < (PC) + 2$, $(Rn) < (Rn) - 1$, якщо $(Rn) = 0$, то $(PC) < (PC) + rel$
DJNZ ad, rel	2		$(PC) < (PC) + 2$, $(ad) < (ad) - 1$, якщо $(ad) = 0$, то $(PC) < (PC) + rel$
CJNE A, ad, rel	3		$(PC) < (PC) + 3$, якщо $(A) = (ad)$, то $(PC) < (PC) + rel$, якщо $(A) < (ad)$, то $(C) < 1$, інакше $(C) < 0$
CJNE A, #d, rel	3		$(PC) < (PC) + 3$, якщо $(A) = \#d$, то $(PC) < (PC) + rel$, якщо $(A) < \#d$, то $(C) < 1$, інакше $(C) < 0$
CJNE Rn, #d, rel	3		$(PC) < (PC) + 3$, якщо $(Rn) = \#d$, то $(PC) < (PC) + rel$, якщо $(Rn) < \#d$, то $(C) < 1$, інакше $(C) < 0$
CJNE @Ri, #d, rel	3		$(PC) < (PC) + 3$, якщо $((Ri)) = \#d$, то $(PC) < (PC) + rel$, якщо $((Ri)) < \#d$, то $(C) < 1$, інакше $(C) < 0$
LCALL ad 16	3		$(PC) < (PC) + 3$, $(SP) < (SP) + 1$, $((SP)) < (PC_{0-7})$, $(SP) < (SP) + 1$, $((SP)) < (PC_{8-15})$, $(PC) < ad\ 16$
ACALL ad 11	2		$(PC) < (PC) + 2$, $(SP) < (SP) + 1$, $((SP)) < (PC_{0-7})$, $(SP) < (SP) + 1$, $((SP)) < (PC_{8-15})$, $(PC_{0-10}) < ad\ 11$
RET	1		$(PC_{8-15}) < ((SP))$, $(SP) < (SP) - 1$, $(PC_{0-7}) < ((SP))$, $(SP) < (SP) - 1$
RETI	1		$(PC_{8-15}) < ((SP))$, $(SP) < (SP) - 1$, $(PC_{0-7}) < ((SP))$, $(SP) < (SP) - 1$

7. Команди операцій зсуву OMEOM сімейства МК51 підтримують операції циклічного зсуву акумулятора:

- замкнуті - RR (вправо) і RL (вліво)
- через біт C - RRC(вправо) і RLC (вліво).

У ЗУПП команди зсувних операцій займають 1 байт і виконуються за 1 машинний цикл.

8. Інші команди.

До інших команд відносяться:

- команда очищення акумулятора CLR A, виконання якої перетворює в 0 вміст акумулятора;
- команда інверсії значення акумулятора CPL A;
- "порожня" операція NOP;
- операція обміну тетрадами в акумуляторі SWAP A;
- операція обміну молодшими тетрадами вмісту акумулятора і непрямо адресованої комірки пам'яті XCHD A,@Ri (молодша тетрада - чотири молодших розряди байта)

Ці команди займають у ЗУПП по одному байту і виконуються за один машинний цикл.

Формування ознак.

У процесі виконання операцій формуються ознаки:

C - перенесення;

AC - допоміжне перенесення (перенесення між тетрадами);
 Z - нульове значення вмісту акумулятора;
 OV - переповнення;
 P - парність.

Для збереження ознак використовується регістр PSW, вміст якого ілюструється (рис.8). Навідміну від інших ознак, OMEOM сімейства МК51 не зберігають ознаки Z, тому він інтерпретується не як ознака нульового результату операції, а як ознака нульового значення вмісту акумулятора.

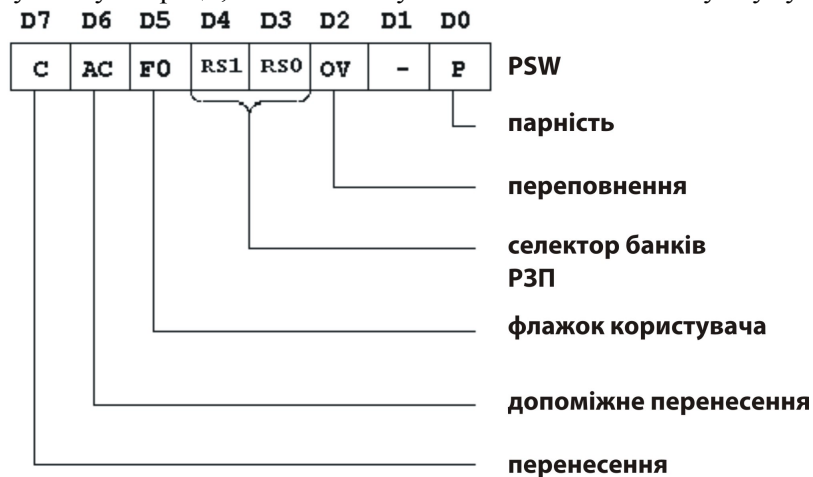


Рис.8

На ознаку C впливає виконання наступних команд: ADD, ADDC, SUBB, MUL, DIV, RLC, RRC, CJNE і бітові операції.

На ознаку OV впливають виконання команд MUL і DIV.

Ознака парності характеризує поточне стан акумулятора, доповнюючи до парного кількість одиниць у його вмісті. Формально займаючи розряд D0 у регістрі ознак PSW, він може бути переданий у стек, однак не підлягає ні встановленню в одиницю, ні скиданню в 0 без зміни вмісту акумулятора.

Наявність у PSW ознаки P дає можливість здійснити по його значенню умовні переходи за допомогою команд:

JB P, <address>

JNB P, <address>

Ознака AC у сукупності з командою DA A призначені для виконання програм, що реалізують роботу процесора в десятковій системі числення.

Розряд D5 у регістрі ознак є флажком користувача F0 і може використовуватися з розсуду програміста; зручність цього флажка полягає в тім, що він зберігається, наприклад, у стеку, разом з іншими необхідними елементами PSW.